

**APPARATUS AND METHOD FOR VERIFYING A LOGIC  
FUNCTION OF A SEMICONDUCTOR CHIP**

5

**CROSS REFERENCES TO RELATED APPLICATIONS**

This application is related to Application Nos.  
09/504,889, filed Feb. 16, 2000 and 09/680,467, filed  
October 6, 2000, both assigned to the assignee of the  
10 present invention.

**FIELD OF THE INVENTION**

The present invention relates generally to emulating a  
logic function of a semiconductor chip, and more  
15 particularly, to verifying a logic function of a  
semiconductor chip through emulation based on a software  
algorithm or model that defines an internal logic design of  
the semiconductor chip.

20

**BACKGROUND OF THE INVENTION**

A successful design of a semiconductor chip  
incorporating an integrated circuit requires ever-increasing  
efforts and time as the complexity of such chips and those  
systems integrating such chips have increased drastically.  
25 Since it is desirable to eliminate the design errors in a  
design procedure before fabrication of such chips,

verification of the logic design of the chip is much needed. Among various methods of verification is an emulation method that tries to verify the functional correctness of a logic function (hereinafter, referred to as "target logic") designed for a semiconductor chip ("target chip") in conjunction with an application system ("target system") in which the target chip will be used.

Traditionally, verification of a logic function has been performed by software or hardware modeling or a combination of the two. Among these, software modeling describes the behavior of the logic function of the target chip through software written in either a high-level programming language such as C/C++ or a hardware description language (HDL) such as VHDL and Verilog. However, HDL does not lend itself to a viable description or debugging of the logic function of the target chip compared to a high-level programming language, and speed of execution of hardware models coded in HDL is very limited. Thus, HDL is known to have disadvantages associated with time spent in designing and verifying the logic function.

For the above reasons, a software programming language, which is easier to code and debug and runs faster compared to HDL, is preferably used in modeling the logic function of the target chip. According to software modeling based on a high-level programming language, the logic function of the target chip is first modeled and the functional behavior of

the logic is verified in conjunction with the surrounding hardware system or a target system. Thereafter, the logic is partitioned manually or possibly with some automatic means into parts to be implemented as hardware and software, 5 respectively. This approach would significantly save the design and verification time, compared to the conventional HDL-based emulation method, and provide more efficient emulation environment for such cases where a logic design becomes extremely complex, especially with the growth of 10 "System-on-a-Chip."

U.S. Patent No. 4,901,259 discloses an "ASIC emulator" which enables software models implemented in a high-level programming language as well as those in HDL to be run on a general-purpose computer connected to a target system via 15 interface. It is basically a generalized pin I/O synchronizer that provides synchronization of pin signals produced by an ASIC software model, which is running on a general-purpose computer, with pin signals produced by hardware in a target system. In other words, all the pin 20 signals for interface with the target system are produced by an ASIC software model, while the ASIC emulator converts the pin signal values into electrical signals in order to send the values to an ASIC bare socket in the target system.

However, when a fast emulation is needed, "ASIC 25 emulator" described in U.S. Patent No. 4,901,259 presents several problems. First, a software model in the host

computer is too slow to execute the whole process related to the generation of pin signals used for interface with external circuits. Second, the communication speed through an I/O port of a full-fledged host computer is generally  
5 very slow as compared with a processor speed. Third, the host computer is too slow to run the algorithm of an ASIC software model because it uses a general-purpose operating system that cannot dedicate its computing resources solely to running the algorithm. Therefore, such ASIC emulator  
10 cannot facilitate a variety of VLSI designs requiring fast emulation.

In order to solve these prior art problems, the inventors of the present invention proposed, in U.S. Patent Application No. 09/680,467, incorporated herein by reference,  
15 an in-circuit emulator which enables fast verification of the logic function of a target chip modeled in a high-level programming language such as C or C++ at a system level where the target chip is coupled to the target system. The emulator comprises a processing engine including at least  
20 one processor that executes a compiled code of an algorithm written in a high-level programming language and a target interface engine including at least one reconfigurable chip, such as Field-Programmable Gate Arrays (FPGAs) or Programmable Logic Devices (PLDs), which generates pin  
25 signals used for external interface with the target system.

With this emulator, however, the processing engine and

the target interface engine may be operating at different processing speeds and a communication delay may occur between the processing engine and the target interface engine due to an interface therebetween. In other words, a  
5 change in a software variable during execution of the compiled algorithm may not necessarily correspond to pin signals generated at the time of the change in the software variable. In order to properly debug a chip algorithm in an emulating environment as disclosed in copending Application  
10 No. 09/680,467 or in equivalent environments, it is required to monitor pin signals generated by the target interface engine in synchronization with variables of an algorithm executed by the processing engine.

Accordingly, there exists a need in the art to provide  
15 a mechanism which enables hardware pin signals generated by the target interface engine to be monitored in synchronization with variables of an algorithm executed by the processing engine in a logic chip emulation environment where the processing engine and the target interface engine  
20 interact with each other.

#### SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide an improved apparatus and method for verifying a  
25 logic function of a target chip which enables easy and fast debugging of the logic design of the target chip.

It is another object of the present invention to provide an improved apparatus and method for verifying a logic function of a target chip which facilitates tracking time-varying changes in software variables of the algorithm  
5 executed by a processing engine and time-varying changes in pin signals generated by a target interface engine, in a logic chip emulation environment where the processing engine and the target interface engine interact with each other.

It is still another object of the present invention to  
10 provide an improved apparatus and method for verifying a logic function of a target chip which enables hardware pin signals generated by a target interface engine to be monitored in synchronization with software variables of an algorithm executed by a processing engine, in a logic chip  
15 emulation environment where the processing engine and the target interface engine interact with each other.

According to one aspect of the present invention, an emulator for verifying a logic design of a target chip to be mounted in a target system is provided, which emulator  
20 comprises a processing engine for executing a software algorithm corresponding to the logic design of the target chip, and a target interface engine interfacing with the target system for transmitting/receiving pin signals to/from the target system in response to the execution of the  
25 software algorithm by the processing engine. Within the software algorithm, there are defined one or more software

variables. The software variables and the pin signals are time-variant with the execution of the algorithm. The processing engine comprises means for finding correspondence between the software variables and the pin signals at a predetermined time.

According to another embodiment of the present invention, the target interface engine comprises a pin signal generator and a pin signal monitor. The pin signal generator generates pin signals to be transmitted to the target system and receives pin signals from the target system. The pin signal monitor receives and stores the pin signals communicated between the pin signal generator and the target system. The pin signal monitor detects an occurrence of a first trigger event defined by a predetermined combination of the values of the pin signals, and transmits a first interrupt request signal to the processing engine, in response to the occurrence of the first trigger event. Upon receiving the first interrupt request signal, the processing engine copies pin signals, which have been stored in the pin signal monitor, to a pin signal buffer.

According to yet another embodiment of the present invention, the processing engine comprises a software variable monitor for monitoring the time-varying software variable and controlling the values of the monitored software variables to be stored. The software variable

monitor detects an occurrence of a second trigger event defined by a predetermined combination of the values for the time-varying software variables, and outputs a second interrupt request signal in response to the occurrence of the second trigger event. The processing engine copies pin signals, which have been stored in the pin signal monitor, to the pin signal buffer in response to the second interrupt request signal.

According to the foregoing embodiments of the present invention, information loss due to communication time delay between the processing engine and the target interface engine can be compensated. Therefore, the values of the software variables and the values of the hardware pin signals corresponding in time thereto can be monitored in synchronization with each other, and as a result, the verification of the logic function of the target chip can be made easier.

#### BRIEF DESCRIPTION OF THE DRAWINGS

The features of the present invention which are believed to be novel are set forth with particularity in the appended claims. The present invention, both as to its organization and manner of operation, together with further objects and advantages thereof, may best be understood with reference to the following description, taken in conjunction with the accompanying drawings in which:



FIG. 1 shows a configuration of an overall system where an emulator according to the present invention is coupled to a target system;

FIG. 2 shows a detailed configuration of a processing engine of an emulator as shown in FIG. 1;

FIG. 3 shows a detailed configuration of a target interface engine of an emulator as shown in FIG. 1;

FIG. 4 shows a configuration of a memory in a processing engine as shown in FIG. 2;

FIG. 5 shows a detailed configuration of a pin signal monitor;

FIG. 6 is a view to explain a process of selecting pin signals of interest out of a plurality of pin signals by a signal selector;

FIG. 7 shows a detailed configuration of a signal sampler;

FIG. 8 shows a detailed configuration of a trigger event detector;

FIG. 9 is a flow chart explaining a series of steps to be taken by a pin signal monitor responsive to an occurrence of a hardware trigger event;

FIG. 10 is a flow chart explaining a series of steps to be taken by a processing engine upon detection of an occurrence of a hardware trigger event by a pin signal monitor;

FIG. 11 is a flow chart explaining a series of steps

to be taken by a processing engine responsive to an occurrence of a software trigger event;

FIG. 12 is a timing diagram illustrating a timed relationship between monitored values of software variables  
5 and pin signals in a conventional emulator; and

FIG. 13 is a view to illustrate how to keep track of changes in a software variable to prevent loss thereof according to the present invention.

#### 10 DETAILED DESCRIPTION OF THE PRESENT INVENTION

Referring to FIG. 1, there is shown an overall system where an emulator according to the present invention and a target system are coupled to each other. The emulator 100 generally comprises a processing engine 101, a target  
15 interface engine 102, and a communication channel 103 for interfacing these engines with each other such as 33MHz PCI bus or Accelerated Graphics Port (AGP). The processing engine 101 communicates with a host computer 110 through a communication port 105 such as RS-232C or USB. The target  
20 interface engine 102 is connected to a socket 121 of the target system 120, on which a target chip will be mounted eventually, through a socket adapter 104. The target chip mounted on the socket 121 is connected to other logic 123 or an integrated circuit 124 of the target system 120 to  
25 execute at least a part of the logic functions of the target system 120. As shown in FIG. 1, a target system 120 may

have a plurality of target chips to be emulated. In this case, another target chip may be connected to another emulator 130 through another socket 122 of the target system 120.

5           FIG. 2 shows one detailed embodiment of a processing engine in FIG. 1. The processing engine 101 includes a processor 201, a memory 202, a host bridge 203, a timer 204, and a miscellaneous logic 205. A software algorithm defining a logic function of the target chip and described  
10 in a high level language such as C/C++ or its equivalent is compiled by a host computer to an executable code. The compiled code is then downloaded through a communication port 105 and stored to the memory 202. The processor 201 in the processing engine 101 performs the application-specific  
15 internal logic function of the target chip such as signal processing, modulation/demodulation, encryption/decryption by executing the downloaded code. The processor 201 and memory 202 are connected to the target interface engine 102, which will be described later, via the host bridge 203. The  
20 miscellaneous logic 205 includes a communication port 105 and additional circuits, such as a clock generator, a voltage converter, etc. (not shown), that are needed for the processing engine 101 to operate. The timer 204 provides  
25 monitoring of pin signals and software variables, also to be explained later, and may be implemented by a counter.

FIG. 4 shows a configuration of a memory in a processing engine as shown in FIG. 2 according to one embodiment of the present invention. The memory 202 is partitioned into several areas storing programs or data such as a target chip algorithm 400, a software variable monitor 401, an interrupt handler 402, a target interface engine controller 403, a software variable buffer 404, a pin signal buffer 405, and a variable/signal analyzer 406. The target chip algorithm 400 stores downloaded executable code relating to the internal function of the target chip. The software variable monitor 401 stores functional routines which will be inserted at the locations of the source code of the target chip algorithm where software variables to be monitored are expected to change their values. Besides the executable code for the target chip algorithm, the functional routines are also downloaded and stored in a compiled form by the host computer 110.

In case that a variable "A" is desired to be monitored in the target chip algorithm 400, the functional routine is added to or changes portions of the source code relating to the variable "A." For example, when an instruction of "A = b;" is found in the source code, that portion of the program is modified to a software routine of "SW\_MONITOR(&A, b);." In one embodiment, the routine "SW\_MONITOR(&A, b);" may not only assign value b to the variable "A" but also storing the value of the variable "A" as changed in the software

variable buffer 404 of the memory 202. Therefore, when the processor 201 comes across this type of functional routine during the execution of the target chip algorithm 400, the changes of the variables monitored by the routine get stored  
5 in the software variable buffer 404. In this case, the software variable buffer 404 may also store timing information associated with the variable change by referencing the timer 204 (FIG. 2). In addition, in the software variable monitor 401, a user-defined software  
10 trigger condition, represented by a combination of selected variables in the target chip algorithm 400, may be set such that when the trigger condition is met, a software interrupt takes place. The software interrupt is handled by the interrupt handler 402, to be described next.

15 The interrupt handler 402 stores software routines that respond and handle interrupt requests occurring while the target chip algorithm 400 is executed by the processing engine 101. For example, it handles an interrupt that is generated when the values of software variables being  
20 monitored and stored in the software variable buffer 404 by the software variable monitor 401 coincide with a predetermined software trigger condition, as explained above. In addition to the software interrupt, it responds to a hardware interrupt from the target interface engine 102  
25 defined by a combination of the values of pin signals. Further details of the interrupt handler 402 are discussed

below.

The target interface engine controller 403 stores routines related to the control of the target interface engine 102. In response to the interrupt handling by the  
5 interrupt handler 402, the target interface engine controller 403 reads the values of pin signals monitored and stored by the pin signal monitor 302 of the target interface engine 102 and then stores them in the pin signal buffer 405 of the memory 202. The variable/signal analyzer 406, using  
10 information stored in the software variable buffer 404 and the pin signal buffer 405, analyzes the progress or error during emulation.

FIG. 3 illustrates an embodiment of a target interface engine according to the present invention. The target  
15 interface engine 102 comprises a communication channel controller 300, a pin signal generator 301 and a pin signal monitor 302. The pin signal generator 301 and the pin signal monitor 302 communicate with the processing engine 101 via a communication channel 103 under control of the  
20 communication channel controller 300. The pin signal generator 301 transmits/receives external pin signals such as read, write and control signals to/from the target system 120 in response to the execution of the target chip algorithm 400 by the processing engine 101. The pin signal  
25 generator 301 is a kind of logic device that is properly designed depending on the application of the target system

120 so that it communicates with logic circuits and/or integrated circuits of the target system 120. In one embodiment of the present invention, the pin signal generator 301 may be embodied in a programmable/reconfigurable chip, such as FPGA or PLD. Because the pin signal generator 301 provides the external pin signals 104 to the target system 120 as if a real chip would have, the target system 120 can properly operate as designed.

10 The pin signal monitor 302 receives as input the pin signals 304 being communicated between the pin signal generator 301 and the target system 120, and/or the internal pin signals 303 of the pin signal generator 301. The pin signal monitor 302 may also be embodied in a programmable/reconfigurable chip such as FPGA or PLD. As shown in more detail in FIG 5, the pin signal monitor 302 comprises a signal selector 501, a signal sampler 502, a trigger event detector 503, a buffer 504, a clock generator 506, a stamper 507, an acquisition memory 505 and a pin signal monitor controller 500. The signal selector 501 receives as input interface signals 304 between the pin signal generator 301 and the target system 120, and the internal signals 303 of the pin signal generator 301 for selecting signals that are predetermined by the user to be monitored. In other words, the signal selector 501 functions to select part of pin signals to be monitored

among signals that are input to the pin signal monitor 302. What signals would be selected is decided by the user and thus may be changed according to the user's need.

FIG. 6 schematically shows an embodiment of a signal selector which selects out three signals from 5 input signals. The user may program a reconfigurable chip such as FPGA or PLD by selectively opening/closing a plurality of connections pre-defined on a programmable area 603 of the reconfigurable chip so as to select desired input signals. Specifically, FIG. 6 shows that output lines O1, O2 and O3 are respectively coupled to input lines I4, I2 and I3. By using this kind of selector, the user can select pin signals of interest and in a desired order.

Referring back to FIG. 5, pin signals selected by the signal selector 501 are sampled by the signal sampler 502 based on a clock signal generated by the clock generator 506. As shown in FIG. 7, the signal sampler 502 comprises a plurality of flip-flops 701a-701n, where n is the number of pin signals 508 selected by the pin signal selector 501. Each flip-flop, synchronized to the clock signal of the clock generator 506, samples input pin signals 508 to produce output signals 509. In order to discriminate sampling times of sampled signals, timing information 511 generated by the stamper 507 is also included in the output signals 509 from the signal sampler 502. Preferably, a simple counter may be used as the stamper 507, which begins



to count when the emulator is initialized and counts up in response to the clock signal.

As shown in FIG. 5, the sampled pin signals 509 along with time information are temporarily stored in the buffer 504 before finally stored in the acquisition memory 505. In this way, the acquisition memory 505 stores time/stamp information as well as sampled pin signals. Therefore, it is possible to accurately locate pin signals at the time of a particular trigger event in the acquisition memory 505 by referencing stamp information 808 (see FIG. 8) which has been output at the time of that particular trigger event from a trigger event detector 503, as will be explained later, and has been stored in the memory 505 along with the associated pin signals. The buffer 504 is provided to compensate a difference between the sampling speed of the signal sampler 502 and the access speed of the acquisition memory 505. The buffer 504 may further comprise a compression function to enable the memory 505 to store more pin signal information. Preferably, the memory 505 has a circular first-in-first-out (FIFO) structure such that pin signals from the buffer 504 are sequentially stored and the oldest data is deleted to accommodate new pin signals. The memory 505 may preferably be implemented by a high-speed memory such as SRAM and may have expandability.

The sampled pin signals by the signal sampler 502 are also input to the trigger event detector 503. The detector

503 determines whether a predetermined combination of the sampled signals coincides with a trigger condition set by the user. If so, the detector 503 generates an interrupt request signal.

5        FIG. 8 shows an embodiment of a trigger event detector 503. The detector 503 comprises a plurality of trigger unit blocks 800a, 800b that produce event signals 805a, 805b, and a trigger controller 806 that outputs a trigger signal 807 when the sequence of event signals from the trigger unit  
10    blocks is judged to match a predetermined sequence. Each of the trigger unit blocks 800a, 800b includes a mask register 801a for designating the locations of pin signals used in a trigger condition, a pattern register 802a for designating a pattern of values for the pin signals defined as the trigger  
15    condition, and a plurality of logic gates (XOR, AND gates) for comparing the contents of the registers 801a, 802a with the sampled data 509 at the bit level. Data in the mask register 801a and pattern register 802a, used as a trigger condition are predetermined as desired by the user under  
20    control of the pin signal monitor controller 500. In this case, it should be noted that although the sampled pin signals and the contents of the pattern register 802a are compared at the bit level, the comparison results at all bit positions are not directly used for generating the event  
25    signal 805a. That is, the comparison results only at bit positions where bits of the mask register 801a are set to 1

are used for generating the event signal 805a.

Therefore, this embodiment allows certain pin signals among sampled pin signals to be used as a trigger condition. However, a trigger event detector according to the present  
5 invention is not limited to what was described above. It can have only a pattern register 802a so that the comparison results between the sampled pin signals and the contents of the pattern register at all bit positions can be used to produce an event signal. Moreover, each trigger unit block  
10 800a, 800b may be designed to include a presettable down-counter 803a so that the event signal 805a is not generated until the number of pattern-matching events equals to the preset value of the counter 803a.

Event signals 805a, 805b generated by the trigger unit  
15 blocks 800a, 800b are input to the trigger controller 806, which determines whether the sequence of the event signals 805a, 805b matches a predetermined one and produces a final trigger signal 807 when they match. The setting of a desired sequence for the event signals can be achieved by  
20 programming a state machine in the trigger controller 806.

The trigger event detector 503, as shown in FIGs. 5 and 8, is provided with stamp information from the stamper 507 as well as the sampled pin signals 509. Thus, the trigger event detector 503 permits the processing engine 103  
25 to precisely identify the time of occurrence of the trigger event by providing stamp information 808 at the time of the

final trigger signal 807. When a trigger event occurs, the trigger signal 807 and associated stamp information 808 generated by the controller 806 are provided to the pin signal monitor controller 500, which in turn, sends an interrupt to the processing engine 101 through the communication channel controller 300. Interrupt handling by the processing engine 101 will be described later.

Referring back to FIG. 5, the pin signal monitor controller 500 of the pin signal monitor 302 communicates with the processing engine 101 through the communication channel controller 300 and controls each of the above-mentioned function blocks within the pin signal monitor 302. Specifically, the pin signal monitor controller 500 controls the signal selector 501 to designate certain pin signals that would be selected among a plurality of pin signals input to the pin signal monitor 302. The pin signal monitor controller 500 also controls the clock generator 506 and the stamper 507 by using control signals 512, 513 in order to control sampling operation of the signal sampler 502. Further, the pin signal monitor controller 500 controls the trigger event detector 503 for setting up a trigger condition, and controls the acquisition memory 505 for delivering stored pin signal information to the processing engine 101 through the communication channel controller 300.

FIGs. 9-11 are, respectively, flow charts explaining a series of steps to be taken in response to a hardware

trigger detected at the pin signal monitor 302 of the target interface engine 102, and a software trigger at the software variable monitor 401 of the processing engine 101.

Referring to FIG. 9, the process at the pin signal  
5 monitor 302 in response to a hardware trigger event detected by the trigger event detector 503 of the pin signal monitor 302 will be explained. First, the emulation system is initialized at step 901. Then, at step 902, the stamper 507 inside of the pin signal monitor 302 is initialized in  
10 synchronization with the timer 204 of the processing engine 101. At step 903, it is determined whether sampling of pin signals should begin. If so, at step 904, some of input/output pin signals 304 between the pin signal generator 301 and the target system 120, and internal pin  
15 signals 303 of the pin signal generator 301 are selected by the signal selector 501. The selected signals are sampled by the signal sampler 502 at rate of a clock signal provided by the clock generator 506 and stored in the buffer 504 before forwarded to the acquisition memory 505. After the  
20 sampling step 904, the sampled pin signals are examined, at step 905, by the trigger event detector 503 to see if they meet a predetermined trigger condition. If no trigger event is detected, the sampling and storing step 904 is repeated. Otherwise, the sampling is suspended at step 906 and an  
25 interrupt request to the processing engine 101 is generated at step 907. In response to the interrupt request, the

processing engine 101 analyzes the values of software variables and pin signals in question and sends a control signal to reactivate the pin signal monitor 302 after the analysis. At step 908, it is determined whether the  
5 reactivating control signal was received and, if so, returning to step 903, the pin signal monitor 302 resumes sampling of pin signals.

FIG. 10 illustrates a series of steps to be taken by the processing engine 101 in response to an interrupt  
10 request that follows from the detection of a hardware trigger event at the pin signal monitor 302. First, after the initialization of the emulation system at step 1000, the processor 201 in the processing engine 101 begins to execute the target chip algorithm 400 stored in the memory 202. At  
15 step 1001, the processor 201 determines whether an interrupt has been requested from the pin signal monitor 302. If positive, at step 1002, pin signal information stored in the acquisition memory 505 of the pin signal monitor 302 is copied to the pin signal buffer section 405 of the memory  
20 202. At step 1003, the processor 201 activates the variable/signal analyzer 406 within the memory 202 to perform debugging using the contents of the software variable buffer 404 and pin signal buffer 405. Lastly, at  
25 step 1004, the processor 201 transmits a reactivating signal to the pin signal monitor 302 to be ready for a next trigger event.

FIG. 11 illustrates a series of steps to be taken by the processing engine 101 in response to a software trigger event which may occur during the execution of the target chip algorithm 400. After the initialization of the emulation system at step 1101, the timer 204 is initialized at step 1102 and, at the same time, the stamper 507 of the pin signal monitor 302 is also initialized in synchronization with the timer 204 as explained above with reference to FIG. 9. At step 1103, the processor 201 begins to execute the target chip algorithm 400. If it is determined that the end of the algorithm is reached at step 1104, the process proceeds to step 1105 and is finished. On the other hand, if it is determined at step 1104 that the execution of the algorithm is yet not finished, the process proceeds to step 1106 where it is checked whether the processor 201 encounters a software variable monitor routine in the target chip algorithm 400. If determined that no software variable monitor routine appears at step 1106, returning to step 1103, the processor 201 proceeds with the remaining portion of the algorithm 400 and continues until a software variable routine is encountered. Otherwise, if determined that a software variable monitor routine is encountered at step 1106, the processor 201, at step 1107, executes the monitoring routine and stores the values of the software variables as monitored by the monitoring routine in the software variable buffer 404 of the memory 202. At step

1108, it is checked whether a software trigger event has occurred, based on the values of the software variables as monitored by the monitoring routine. In case of no trigger event, the processor 201 continues to execute the remaining  
5 portions of the target chip algorithm 400. If the values of the monitored software variables coincide with those predefined as a trigger condition, the sampling operation for pin signals by the pin signal monitor 302 is suspended and the pin signal information stored in the acquisition  
10 memory 505 of the pin signal monitor 302 at the event time is read and copied to the pin signal buffer 405 at step 1110. Finally, at step 1111, the processor 201 executes the variable/signal analyzer 406 within the memory 202 to perform debugging, using the contents of the software  
15 variable buffer 404 and pin signal buffer 405. After debugging, the process returns to step 1103 and the remaining portions of the algorithm 400 continue to be executed.

Next will be explained the way pin signals and  
20 software variables that are monitored at the processing engine and at the target interface engine, respectively, are saved without loss according to the present invention.

Referring to FIG. 12, there is shown a timing diagram illustrating a timed relationship between values of software  
25 variables monitored by the processing engine and hardware pin signals by the target interface engine in a conventional



emulator. During the emulation, if a trigger event 1201 predefined by the user as a combination of pin signals occurs, this event is detected by the trigger event detector of the pin signal monitor, and a trigger interrupt request signal 1202 based on the detection of the trigger event is transferred to the processor of the processing engine. At the same time, the pin signal monitor begins to store all the changes 1203 in pin signals monitored after the occurrence of the trigger event in the pin signal memory.

On the other hand, the processing engine does not begin to store the changes 1206 in software variables associated with the hardware trigger event in the software variable buffer until the timer for synchronization of software variables and hardware pin signals gets started after the trigger request signal 1202 is sent to the processor and an acknowledgement signal 1204 thereto is transmitted by the processor. Here, however, transmission of the interrupt request signal 1202 necessarily takes time. Moreover, even after the interrupt request signal 1202 reaches the processor, the processor cannot immediately handle the interrupt request if it was in the middle of performing a software variable monitor routine inserted in the target chip algorithm. The processor can acknowledge to the interrupt request signal 1202 only after it completes all software variable monitor routines being executed. Because of the interrupt-wait time incurred by these time delays as

described, any changes in software variables during this interrupt-wait time cannot be stored. In reality, in case that the pin signal monitor and the processing engine are coupled via a PCI bus as a communication channel, the interrupt-wait time could amount to several hundreds of or, in a worst scenario, several thousands of PCI clock cycles. As a result, data loss would be disastrous.

FIG. 13 illustrates how to keep track of changes in software variables in response to a hardware trigger event detected by the pin signal monitor in accordance with the present invention. The horizontal axis represents time, and circles and squares on the time axis represent changes in software variables. Horizontally extending rectangles below the time axis represent a variable recording window showing contents stored in the software variable buffer at each timing, i.e., the values of the software variables as changed prior to each timing. Specifically, window A shows contents of the software variable buffer storing the values of the software variables as changed prior to time A. Likewise, windows B and C show contents of the software variable buffer storing the values of the software variables as changed prior to times B and C, respectively. Here, the capacity of the software variable buffer, i.e., the size of windows A to C, is preferable to be such that it could record all the changes in software variables during the above described interrupt-wait time. As such, in accordance

with the present invention, every time there is a change in a software variable (i.e., when a software variable monitor routine is executed), the change gets stored and updated in the software variable buffer at least during a period of  
5 time corresponding to the interrupt-wait time. Thus, any changes in the software variables during the intervening period between the detection of a trigger event 1301 and an acknowledgement thereto 1302 can be stored without loss.

Although the above description was devoted to explain  
10 a delay in monitoring software variables in case of the detection of a hardware trigger event and a solution thereto in accordance with the present invention, it should be noted that the description also applies in the same manner to a delay in monitoring hardware pin signals in case of the  
15 detection of a software trigger event. Therefore, according to the present invention, the loss of hardware pin signals during the delay period is also prevented.

As described in the foregoing, according to the present invention, in a logic chip emulation environment  
20 where the processing engine and the target interface engine interact with each other, hardware pin signals generated by the target interface engine can be monitored in synchronization with software variables within an algorithm executed by the processing engine, and therefore, total  
25 emulation time for an overall emulation system including a target system can be reduced.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that changes and modifications may be made without departing from this invention in its broader aspects and, therefore, the aim in the appended claims is to cover all such changes and modifications, as fall within the true spirit and scope of this invention.

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232